

## SYSTEMS AND METHODS FOR PERFORMING DATA COLLECTION

### BACKGROUND OF THE INVENTION

[0001] Various processes and systems are known to effect data entry within the insurance industry. Typically such data entry might include a person reviewing a hand written form and transferring the information on the handwritten form into electronic form, i.e., by entering the information into a database using a computer, for example. The various systems that are known contain various pieces of functionality.

[0002] However, none of the known systems provide all or the type of functionality that is required in today's competitive market. A data entry person working for an insurance related entity is faced with complex forms from which to retrieve information. The data entry process is often complicated by the manner in which the form was completed by the applicant, especially if the paper form was not specifically designed with data entry in mind. For example, the applicant may not provide complete information, may enter remarks on a margin of a form, may fill in two selections where the question requires a "one or the other" answer or may materially alter entries. Further, when faced with hundreds of forms to complete, every efficiency that can be provided to the data entry person is highly desired.

[0003] Accordingly, the systems and methods of the embodiments of the invention provide a novel arrangement of features that address the above needs, as well as others.

### BRIEF SUMMARY OF THE INVENTION

[0004] Systems and methods are provided for performing data collection of insurance related submitted data, provided by an applicant for insurance, on a submitted form. In accordance with one embodiment of the invention, the system may comprise a data input portion that inputs the

submitted form so that the submitted data on the submitted form is available to a data entry operator; and a data collector tool, which processes entered data that is entered by the data entry operator, onto an internal electronic form, based on the submitted data submitted by the applicant. The data collector may include a form flow portion that presents the data entry operator with a series of form flows for collection of the submitted data, the form flows progressing through various user interface screens in collection of the submitted data, the form flows using content sensitive logic; and a metadata portion that generates metadata, the metadata containing information about the entered data, the metadata being progressively generated during progression of the form flows through the various user interface screens.

## 10 BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The present invention can be more fully understood by reading the following detailed description together with the accompanying drawings, in which like reference indicators are used to designate like elements, and in which:

15 [0006] Figure 1 is a flowchart showing a high level process in accordance with one aspect of the invention;

[0007] Figure 2 is a block diagram showing an insurance data processing system in accordance with one embodiment of the invention;

[0008] Figure 3 is a block diagram showing a data collector tool in accordance with one embodiment of the invention;

20 [0009] Figure 4 shows examples of various questions in accordance with one aspect of the invention;

[0010] Figure 5 is a user interface showing a component tool in accordance with one aspect of the invention;

[0011] Figure 6 is a table showing navigation buttons that may be provided on a data entry page in accordance with one embodiment of the invention;

[0012] Figure 7 is a user interface showing a basic example of manipulation of form navigation buttons in accordance with one embodiment of the invention;

5 [0013] Figure 8 is a table showing examples of content sensitive logic in accordance with one embodiment of the invention;

[0014] Figure 9 is a user interface showing an example of a question with a “remarks” flow in accordance with one embodiment of the invention;

[0015] Figure 10 shows a sample remarks page in accordance with one embodiment of the  
10 invention;

[0016] Figure 11 shows a sample repeating question in accordance with one embodiment of the invention;

[0017] Figure 12 illustrates an illustrative screen layout in accordance with one embodiment of the invention;

15 [0018] Figure 13 is a user interface showing examples of both a detailed and a normal signature verification in accordance with one embodiment of the invention;

[0019] Figure 14 shows various attributes, i.e., metadata, that may be used in operation of the data collector tool in accordance with one embodiment of the invention;

[0020] Figure 15 is a table that describes conditions in which a validation query is presented  
20 to the data entry operator or other user in accordance with one embodiment of the invention;

[0021] Figure 16 is a user interface that shows a generic mask error query to a data entry operator in accordance with one embodiment of the invention;

[0022] Figure 17 is a table that lists illustrative masks, which may be used in operation of the data collector tool in accordance with one embodiment of the invention;

[0023] Figure 18 is a table that describes a particular condition that a field has “failed the checkbox combination” and the resulting “failed combination” validation window in accordance with one embodiment of the invention;

[0024] Figure 19 is a user interface showing a page level validation screen in accordance with one embodiment of the invention;

[0025] Figure 20 is a sample form flow showing a blend of possible information found on a form flow, as well as descriptive text highlighting the features of the form in accordance with one embodiment of the invention;

[0026] Figure 21 is a flow chart showing a form flow process in accordance with one embodiment of the invention;

[0027] Figure 22 is a flowchart showing further details of “the data collector tool checks items as data entry is performed” step of Figure 21 in accordance with one embodiment of the invention; and

[0028] Figure 23 is a flowchart showing further details of the “data entry operator reviews entered data to determine if the entered data is proper” step of Figure 21 in accordance with one embodiment of the invention.

#### DETAILED DESCRIPTION OF THE INVENTION

[0029] Hereinafter, aspects of a data collector in accordance with various embodiments of the invention will be described. As used herein, any term in the singular may be interpreted to be in the plural, and alternatively, any term in the plural may be interpreted to be in the singular.

The various embodiments of the invention relate to systems and processes to collect data from a source of information, such as a paper document or other medium.

[0030] The invention provides a data collector that allows the transfer of information from a source of information, such as a paper document, to data. In accordance with one aspect of the invention, information relating to insurance, e.g., such as an application for insurance, is submitted for processing. This information is submitted via some medium, such as a paper document. The information will typically be entered into a computer system by a humane being, who might be characterized as a “data collector” or “data entry operator,” for example. In the normal course of business, as should be appreciated, the data entry operator may very well be the only human being to ever physically examine the data. As a result, it is essential that the data collector allow the user, i.e., some other person later relying on the data, to identify missing, altered and/or unreadable pieces of information, for example, such that an automated system could react to that information. In addition, an effective system needs to prevent data entry errors “at the time of data entry” to the maximum extent.

[0031] To this end, the present inventors herein describe systems and methods to effectively input and manipulate data using a “form flow” approach and metadata. Illustratively, a variety of functionality is provided therewith including content sensitive logic features, field level mask capability and data validation function, for example. The provided system forces the user to enter only the necessary information and in the correct format. Finally, since data entry is often performed remotely using a client-server application, the system provides needed performance and response, which is typically very important in the selection of a data entry solution. Accordingly, a technical effect for the disclosed inventive technology is to provide systems and

methods for performing a data entry process, as set forth in the Brief Summary of the Invention, above.

[0032] The invention embodies the concept of a “formflow”. Such approach, as described herein, is in contrast to using an exact digital replication of a paper document to serve as a data collector, which is unsatisfactory to fully and efficiently capture information, as determined by the inventors. Instead, the formflow concept described herein allows free and logical navigation of the data collector to facilitate data entry. Essentially, the formflow serves as a digital “script” that guides the DEO, for example, through the data entry process in a predetermined manner, which is deemed by the designers of the script to be the most logical and/or efficient manner.

Content sensitive logic is an integral part of the formflow concept. That is, certain data elements are presented to the DEO only when appropriate and relevant. Again, this logic is built into the formflow “script” to reduce screen clutter, eliminate distractions to the DEO, and to reduce opportunities for human error. Content sensitive logic features as described herein might be characterized as a feature that progressively discloses further fields (for the data entry operator to complete) based on the content of a field currently displayed.

[0033] In accordance with some aspects of the invention, a data collector is provided in the form of software. The data collector is used by a data entry operator to enter information from a source medium. The source medium might typically be a filled in document, which was completed by an applicant for insurance. However, the source medium might take on other forms, such a human verbally collecting information, for example. The document from which the information is taken might be a life insurance application or related document. However, this invention is not limited to just life insurance applications, but rather is directed to the collection of various types of insurance related data.

[0034] In accordance with various embodiments as described herein, the invention provides a variety of significant features, in conjunction with novel interrelationship between these features. One feature of some embodiments of the invention includes content sensitive logic, as noted above. The content sensitive logic functionality provides context sensitive logic, which examines what the data entry operator enters. The context sensitive logic, based on the particular input, presents one or a whole set of additional questions, for example. Further details of the content sensitive logic feature are described below.

[0035] Further, in the invention, the content sensitive features are utilized in a novel manner in combination with the capturing of metadata. In the capturing of metadata, the system assigns characteristics about the data that the data entry operator enters. For example the data entry operator might see that something has been altered on the original document, which they are keying data for. The data entry operator would still type in the altered information to the system. However, the data entry operator will make a notation that the data was altered. This notation allows the system to take action on this fact during later processing, as is necessary or desired. For example, there may well be significant legal implications if an altered signature or an altered medical detail would have changed the outcome of a particular case. In accordance with one embodiment of the invention, the invention allows this logic to be completely configurable to alert any number of people( i.e., such as a person or team) and/or result in a certain action, such as requesting a follow-up document, for example.

[0036] Figure 1 is a flow chart showing the high-level process in accordance with one embodiment of the invention. The process starts with an insurance applicant 10 completing an insurance form, i.e., a submitted form 14, for example. On the submitted form 14, the insurance applicant 10 enters a variety of information. The information that the insurance applicant 14

enters on the submitted form 14 is herein characterized as “submitted data 15.” Thereafter, the insurance applicant 10 submits a completed submitted form 14 to the insurance company or other entity, as is typical. At some point after the insurance applicant 20 submits the submitted form 14, the submitted data 15 is data entered into a suitable insurance data processing system 20 of the insurance entity, as shown in Figure 1. This data entry may be performed by a data entry operator 11, as shown in Figure 1. As used herein, a data entry operator means any person acting in the role of entering the submitted data 15. The data entry operator 11 retrieves various internal electronic forms 16, which are generated by the insurance data processing system 20. One embodiment of the invention uses barcodes on forms 14 that are read by the insurance data processing system 20 in order to retrieve the correct corresponding data collector and eliminate the possibility of data being entered on the wrong data collector. Based on the information in the submitted form 14, the data entry operator 11 enters various entered data 17 into the internal electronic form 16. The generation of the internal electronic forms 16 and features of the data entry are described in detail below. After the data entry operator 11 has entered the entered data 17, the internal electronic form 16 is then ready for further processing, as may be desired, as shown in step 18 of Figure 1.

[0037] As used herein, the internal electronic form 16 means that the form is prepared internally to a particular entity, for example an insurance company. However, such meaning is clearly inclusive of the situation that the insurance company utilizes some affiliate entity to perform the data entry process, as described herein.

[0038] Figure 2 is a block diagram showing the insurance data processing system 20 in further detail, in accordance with one embodiment of the invention. As shown in Figure 2, the insurance data processing system 20 includes a data input portion 22. The data input portion 22



inputs the submitted form 14 into the insurance data processing system 20 and prepares the submitted form 14 for review by the data entry operator 11.

[0039] Further, the insurance data processing system 20 includes a data collector tool 300 and a rules processing portion 24. The data collector tool 300 performs a wide variety of processing associated with data entry as performed by the data entry operator 11, for example. Further, the rules processing portion 24 shows one example of a component that may utilize the entered data 17. That is, the rules processing portion 24 may apply various rules against the entered data 17 so as to prepare an insurance policy, for example.

[0040] In accordance with one embodiment of the invention, Figure 3 is a diagram showing the data collector tool 300 in further detail. The data collector tool 300 includes a form flow portion 310, a metadata portion 320, and a validation portion 330. Further, the data collector tool 300 includes a user interface 302. The illustrative user interface 302 may interface with a data entry operator and/or an exception manager (EM), for example. As used herein, an “exception manager” is a person or machine who deals with submitted data 15 and/or entered data 17, which is deficient in some manner, i.e., such that the internal electronic form 16 needs further attention prior to processing an insurance policy, for example.

[0041] With further reference to Figure 3, the form flow portion 310 performs various form flow processes, as described below. The metadata portion 320 generates various metadata regarding entered data 17, i.e., data that is entered by the data entry operator. Further, the validation portion 330 performs a variety of validation checks as described below. Such validation checks might be performed on a particular field in the internal electronic form 16 or on a page of the internal electronic form 16, for example. The operation of these various components of the data collector tool 300 are further described below.

[0042] In accordance with one aspect of the invention, the invention utilizes various metadata. It is appreciated that there are various types of metadata that the invention may process. One such type of data is altered data. Altered data includes the intentional alteration of content in a document. For example, an applicant “lines-through” an entry, “overwrites” an entry or effects a physical alteration of an application, such as “white-out”, for example.

[0043] Another type of metadata occurs where a field is left intentionally blank.” This “field left intentionally blank” metadata tells the system the data entry operator did not miss entering a field, but that the field was left blank on the original document.

[0044] Another type of metadata indicates that a field is “unreadable.” Metadata indicating an unreadable situation is used when a field has been completed on the paper form but the data entry operator cannot determine what value has been entered on the document, for example.

[0045] Another type of metadata designates “data entry error.” This metadata is assigned to an entry “after” initial data entry by the data entry operator 11, for example. Such metadata is designed to allow a subsequent operator, i.e., a non-data entry operator or data entry operator, to make a correction to a piece of data and to note that the correction was due to data entry error, and not for some other reasons, e.g., such as an alteration by customer request. For example a quality assurance person may notice that there is an erroneous entry from the data entry operator and fix that erroneous entry. . This metadata will allow the system to recognize that the change was not applicant-initiated so as to avoid a subsequent acknowledgement of the change by the applicant (in the Insurance industry, this is known as a “signed amendment on delivery”).

[0046] Another type of metadata is “failed mask (FM) metadata.” This refers to data that should be formatted a certain way, but was not. For example, if the data entry operator (DEO) encounters a 10 digit social security number, such a number is not valid. However, it may be the

situation that a legal requirement mandates that such data be keyed in exactly as the data is set forth on the submitted form 14, e.g., a submitted insurance application. Accordingly, the failed mask option will force the system to accept the keyed information, but will allow the system (or a person working in the system) to take further action later in processing since that data may have a problem. The system will first advise the DEO that the information entered is incorrectly formatted but will allow him to intentionally override the mask in order to accurately represent the data as received on the document.

[0047] Another type of metadata is “page level validation.” Page level validation enforces the metadata requirements. After the DEO has completed a set number of predefined entries, the system will check to ensure that all the required entries either have data or that a “field left intentionally blank” metadata notation has been made. For example, sometimes fields are left blank by the person filling in the form because they either do not know the answer, don’t want to disclose the information or simply just missed the question. If the DEO does not make some sort of notation on an empty field, the system will not allow the data entry operator to submit the document until the data entry operator corrects the situation.

[0048] Hereinafter, further details of the various embodiments of the invention will be described. Figure 21 is a flowchart in accordance with one embodiment of the invention showing further details of the general process shown in Figure 1. The process of Figure 21 may be performed by the data collector tool 300, in accordance with one embodiment of the invention. In accordance with one embodiment of the invention, the system automatically discerns the type of form 14 received from the insurance applicant 10. This might be done using a suitable form number, for example, or other identifying indicators, or associated with, the particular form, e.g., such as a bar code. Then, the system opens a respective data collector for

that particular form. As a result, such processing eliminates the possibility of entering data on a wrong data collector. That is, the data collector tool as described herein might in operation rely on a plurality of programs or a plurality of program code modules. Different modules might be used for different forms, or even for different parts of the same form. Such an arrangement of identifying the submitted form 14 limits the occurrence of a wrong data collector being used for a particular form.

[0049] Thereafter, the Data Entry Operator (DEO) begins keying the information on the submitted form 14 into the corresponding fields on the particular form retrieved, i.e., on the internal electronic form 16, as shown in Figure 1. As the data entry operator 11 types information into the data collector, the data collector tool 300 ( and in particular the validation portion 330) begins checking a variety of items. For example, the data collector tool 300 checks whether the data is properly formatted. The system performs this checking by using field-level “masks,” in accordance with one feature of the invention.

[0050] Also, if the data entry operator observes that there is missing data on the original form, the data entry operator can still enter the other information and effectively complete entry of the data for the particular form. However, as to the information that is incomplete, the data entry operator makes an appropriate metadata notation through operation of the metadata portion 320, i.e., so that another person reviewing the data at a later time may know the nature of the data. This arrangement of course is in contrast to a system that either allows the data entry operator to send through a form with missing data, or alternatively, a system that would not allow the data entry operator to proceed to the next field or section as a result of the missing data.

[0051] With further reference to Figure 21, showing further details of the data entry process, as shown in Figure 21, the data collection process starts in step 400 in which submitted data is

ready for data entry. Then, in step 410, the particular type of form that was received from the applicant is determined. This determination might be performed based on a bar code on the submitted form, for example. After step 410, the process passes to step 420.

[0052] In step 420, the data collector tool opens a respective data collector for that particular form. Then, in step 430, the data entry operator (data entry associate) begins keying the information on the submitted form into corresponding fields on the internal electronic form, i.e., through use of the form flow process, as described herein. That is, in step 430, the progression of steps through which the data entry operator enters data is controlled by a predetermined form flow, as described herein. After step 430, the process of Figure 21 passes to step 440.

[0053] In step 440, the data collector tool checks items as data entry is performed. Further, in step 450, the data entry operator reviews entered data and inputs metadata as appropriate. Further, in step 460, the data collector tool operates to perform content sensitive logic. As can be seen by the iterative loop 470 of Figure 21, the steps (440, 450, 460) are repeatedly performed as is necessary for entry of the data. Note also that the order of the steps (440, 450, 460) may of course vary. After step 460, the process passes to step 490, in which the process ends for that particular form.

[0054] Figure 22 is a flowchart showing illustrative further features of the step 440 in which the data collector tool checks items as data entry is performed, in accordance with one embodiment of the invention. As shown in Figure 22, the subprocess starts in step 440 and passes to step 442. In step 442, the data collector, and specifically the validation portion 330 in accordance with one embodiment of the invention, checks whether data is properly formatted using field level masks. Then in step 444, the data collector performs field level validation. After step 444, the process passes to step 446.

[0055] In step 446, the validation portion 330 of the data collector tool 300 performs page level validation. After step 446, the process passes to step 449. In step 449, the process returns to step 450 of Figure 21.

[0056] Figure 23 is a flowchart showing illustrative features of “the data entry associate reviews entered data and inputs metadata as appropriate” step of Figure 21. After the subprocess starts in step 450, the process passes to step 452. In step 452, the data entry operator enters data based on a submitted form. Then, in step 454, the data entry operator reviews the entered data to determine if the entered data is proper. Then, the process passes to step 456.

[0057] In step 456, the data entry operator illustratively determines that the entered data is not proper, but that the entered data does indeed reflect the submitted data. As a result, in step 457, the data entry operator enters appropriate metadata to reflect the non-proper nature of the entered data. This might be performed by changing the font attributes of labeling (associated with the particular field), for example. After step 457, the process passes to step 458.

[0058] In step 458, the data entry operator proceeds to further submitted data for entry. Note that this progression of step 458 is controlled by the form flow of the particular electronic form, as described herein. After step 458, the process passes to step 459. In step 459, the process returns to step 460 of Figure 21.

[0059] It is be appreciated that, in completion of a particular form, if the data entry operator sends through missing data, such omission may not get noticed until further in the process.

Substantial delays could then be caused at that later date in the processing. Further, without the system of the invention, if the data entry operator is forced to key something, they might make up the information just to get the document through. This could of course create compliance problems. By allowing the data entry operator to send through missing or incorrect information

by noting it with Metadata, we ensure that problem data is addressed by the appropriate individual in a methodical manner.

[0060] In accordance with embodiments of the invention, a content sensitive logic is used, as discussed above with reference to Figure 21. To explain, as the DEO is keying in information, some fields will cause the content sensitive logic to be activated. What this means is that on some fields, based on what is entered, additional questions or actions may be generated or requested, respectively. These questions or actions are fully configurable meaning that they can change depending on the particular content or value of a selected field, for example. One example, in the implementation of the method, is owner beneficiary information. In this example, the data entry operator selects the TYPE of owner of the policy. The type of owner of the policy can be either an individual, business or trust. If the data entry operator selects “individual,” i.e., as a result of the content of the information the data entry operator is entering, the system generates a predefined set of questions. These questions, in this example, relate to an individual owner. On the other hand, if the data entry operator selected “trust”, i.e., as a result of the content of the information the data entry operator is entering, the system would generate a different set of questions since the legal requirements for trust ownership are different.

[0061] The same type of logic could be used on such things as a car loan. If the car owner is an individual, the data collector tool 300 would ask some basic questions about their finances, for example. However, if a business was selected, different and more detailed questions would be required.

[0062] The content sensitive features as described herein eliminates presenting the DEO with every possible question and having the data entry operator decide what needs to be entered. Instead, the data collector tool 300 presents the DEO only with questions that should be

answered, thus reducing the time it takes to data enter information and reducing opportunities for human error by entering information into the wrong fields. This content sensitive logic in conjunction with the use of metadata in the data collector tool 300 provides a powerful data collection tool.

5 [0063] In accordance with one aspect of the invention, as the DEO gets to a predefined break point, the data collector tool 300 will perform page level validation. In this step, the data collector tool 300 ensures all the required data and metadata has been entered. If there are any problems, the data collector tool 300 alerts the DEO to correct the errors. Though data validation could be performed solely at the field level, page level validation was designed specifically to  
10 improve system performance by reducing the frequency of “callouts” to the server.

[0064] When the DEO completes all the data entry, they simply submit the data. The system then performs a final validation check and sends the data to the next processing step. Based on system routing logic, the collected data may then be sent via workflow to a separate Quality Assurance Representative (QAR) or will be passed through the system for automated “back-end”  
15 processing.

[0065] In either case, any changes made to the entered data, would be noted in the metadata information that something has changed, i.e., through a person working with the metadata portion 320. This allows the entry of the data, as well as changes to the data, be fully auditable. Also, before the corrections are made the user is presented with a list of changes about to be  
20 submitted and the user will either approve or not approve the changes. Another embodiment of the invention provides for complete recording of noted data entry errors by DEO for subsequent remediation / training.



[0066] In accordance with one embodiment of the invention, the invention further provides what might be characterized as validation using real-time call-outs. In this embodiment, the data collector tool 300, upon entry of particular data, identifies that data as being able to be validated based on contacting another system and/or database, for example. Illustratively, the system might identify a particular item of information as being an address. Accordingly, in this embodiment, the system contacts, i.e., calls out to, the U.S. Postal Service to verify the address. This processing might be performed by performing a suitable search on the U.S. Postal Service web site or database to determine if a match is found for the address provided in the form. The results of the call-out, i.e., whether a match was found and the data was verified, is then contained in appropriate metadata. Relatedly, the invention may provide an address lookup feature, as is known. That is, the city and state fields may be auto-populated based on zip code.

[0067] In accordance with one embodiment of the invention, the system allows performance tuning or streamlining. In this embodiment, the data collector system is modified to make improvements based on user experiences in production. In one aspect, the progression of questions may be changed. Alternatively, unnecessary fields may be eliminated, for example. Such performance tuning may result from monitoring and recording data entry activity performed by the data entry operator, for example. Attention might be warranted on a field that consistently appears to slow data entry operators. Additionally, navigation within the formflow may be modified to reduce the number of keystrokes in an effort to expedite the data entry process.

[0068] In accordance with one embodiment of the invention, the system performs cross-population of repetitive data. As can be appreciated, there are several fields (on one or more

forms) that appear more than once. The system identifies these fields as being the same and performs a cross-population of these same fields.

[0069] Hereinafter, various embodiments of the systems and methods of the invention will be described in further detail. In particular, aspects of implementations of various embodiments will be described.

[0070] In accordance with one aspect of the invention, in an insurance application process, data from an applicant may well be in the form of a hard copy document. This document, or another medium in which the application information is presented, may typically be considered a legally binding document. This presents a variety of challenges that are addressed by the processing of the data collector tool 300. Illustratively, all information written on the form should be entered into the data entry application exactly as submitted on the document. This is particularly challenging when information is written on the form incorrectly, or additional information is provided that is not typically found on a form. Further, all changes made on the hardcopy form should be noted as an alteration in the data entry application for later consideration. Also, information that is not readable should be noted for later consideration. Further, information missing from certain fields should be noted for later consideration. The metadata as described above is used to represent these various deficiencies. Further, other metadata may be used to represent other deficiencies.

[0071] As noted above, in accordance with one embodiment of the invention, the invention uses what might be characterized as content sensitive logic. The invention also uses “repeats.” Due to the nature of paper form design, content sensitive logic and repeats are necessary to provide the flexibility needed to capture form information. However, extensive use of content sensitive logic and repeats can make form design very difficult and complex. Form flows can

grow exponentially when adding a combination of these two capabilities in a traditional flow layout document. In order to efficiently document the form flows; the invention provides visual form flows that rely primarily on notes and comments to identify the use of content sensitive logic and repeats.

5 [0072] The invention further provides data mapping. That is, data presented in the form flows is mapped to the system. One methodology for identifying fields for mapping purposes has been based on a naming convention using form name, form section, and field User Interface (UI) label. An alternate or additional methodology may be required in the future for mapping form fields. Alternatively, or in addition to, the invention may use a numbering convention for  
10 fields. This may require more document maintenance, but would be beneficial for the purpose of consistently mapping fields.

[0073] The invention provides an arrangement that offers user-friendly forms that will enable the DEO to quickly and efficiently enter data. The DEO populates an electronic form by reading information found on a hardcopy form (or image) and manually typing such information into the  
15 electronic form. The disclosed data collection system facilitates navigation within each form by minimizing “mouse clicks” and maximizing the utilization of “tab” functionality, as is described below so as to allow the DEO to primarily use the keyboard.

[0074] In accordance with another embodiment of the invention, it might be the particular situation, that “all” the submitted data 15 on a submitted form 14 is not required on the internal  
20 electronic form 16. In this situation, another consideration in practice of the invention is the decision of what information needs to entered by the data entry operator. As can be appreciated, a variety of factors might impact this decision of whether data should be entered. In accordance with one aspect of the invention, various rules may be applied against the entered data, i.e., in the

5 further processing of entered insurance data 18, as shown in Figure 1. The data entry operator should of course enter all information that is needed to run these rules. Further, the entered data may well be used to print an insurance policy, for example. The data needed for a complete policy print should also be entered. Further, particular health conditions of a patient, e.g. HIV, may dictate that certain data be entered. As can be appreciated, some form data is required to be entered for legal reasons. On the other hand, some form data is not required for legal reasons and will not be processed in any manner. Accordingly, such information may not be required to be entered in accordance with this particular embodiment of the invention .

[0075] In accordance with a further aspect of the invention, various mechanisms or tools may be used in the operation of the data collector tool 300. Hereinafter, these tools are described. Thereafter, particular implementation of the various tools are described.

[0076] One tool that the invention may be provided with is an auto-population tool. The auto-population tool works off known or previously entered data so as to populate further identical data fields. For example, the data entry operator may retrieve a particular internal form for completion. The internal form is associated with a particular case number. The particular internal form may then be populated based on previously entered data for that case from another form. Alternatively, data that is entered onto an internal form a first time may be auto-populated onto further fields in the form requesting the same data.

[0077] In accordance with a further aspect of the invention, check boxes 42 and radio buttons 44 may be used, as shown in Figure 4. As is known, check boxes are similar to radio buttons, but their selection model, by convention, is different. Any number of check boxes in a group (none, some, or all) can be selected. A group of radio buttons, on the other hand, must have only one button selected. As can be appreciated, various criteria may influence when and how to use

check boxes and radio buttons. Checkboxes might be used in the internal form wherever radio buttons are used on the submitted form. That is, check boxes provide much more flexibility than radio buttons. This added flexibility will enable the DEO to enter the internal electronic form exactly as shown on the submitted form. For example, a data entry operator can select both yes and no, or they can choose to select neither so as to accurately reflect the information set out on the submitted form.

[0078] However, the particular selection of radio buttons by the data entry operator of course affects validation of the internal electronic form. For example, the data entry operator can select both yes and no. However, the Yes and No checkboxes will be considered a group and will not pass validation if both are selected, in accordance with one embodiment of the invention.

Accordingly, by default, checkboxes will be considered a selection within a group. This mocks a radio button grouping, while allowing the DEO to enter multiple checkboxes. However, in some cases, it may be necessary to set checkboxes as multi-select fields.

[0079] Another option is the use of radio buttons by the data entry operator. However, radio buttons result in a forced action for the data entry operator. This may not be desirable because the role of the DEO is to not be limited to a specific selection. Accordingly, in the practice of the invention, radio buttons might only be used for a question in which it is desired to force the DEO to answer only one item. Radio buttons are generally used to force the DEO to make a judgment about the data, not to capture data entered on the form.

[0080] Another option is the use of a “question” format. In accordance with one embodiment of the invention, this format is only used for form questions with on/off or yes/no type answers.

[0081] Further, the systems and methods of the invention might use combination boxes and/or list boxes. Figure 4 shows examples of such combination boxes and/or list boxes. For example, combination boxes 46 might be used for questions where only one value can be selected. Alternatively, list boxes 48 might be used for questions where more than one value can be selected.

[0082] In implementation of combination boxes and list boxes, various guidelines may be utilized for creating values in lists. For example, a blank value may be placed at the top of all lists and set as the default value. A blank value does not set the blank attribute for the field. That is, it is appreciated that there is of course a distinction between the value of a field (e.g. no value) and metadata about a field indicating that the field was left blank on the submitted form 14. Another guideline is to order lists in alphabetical order (including "Other" values). Alternatively, the lists might follow some logical order.

[0083] In accordance with one aspect of the invention, the data collector tool 300 utilizes a tool that might be characterized as a "component." Figure 5 is illustrative of a component tool 50. The component tool might be described as providing "hidden" flows. A component is displayed by pressing a button associated with the component. Examples of components are ADDRESS and SHARE. Once a component, i.e., the details of a component, is displayed the data collector tool 300 should provide the ability to remove the component, similar to remark flows.

[0084] As should be appreciated, various field labels may be used in the practice of the invention, i.e., so as to show the metadata, for example. Various considerations may be taken into account in the use of field labels. The labels may be positioned as desired. For example, it

may be desired to place list box, combination box (i.e., combo box), and text box labels on the left side of the field whenever possible, or where necessary to place such labels above the field.

[0085] Further, various protocols might be used for Radio Button labels and Check Box Labels. For example, check boxes may be created according to how the question appears on the imaged form, i.e., the submitted form. A question label (showing metadata) may be used for groups of radio buttons. In addition, all checkboxes that are considered grouped for validation may be provided with a label (to show metadata about such grouped checkboxes). This label is where metadata elements are displayed. Illustratively, radio button labels may be placed on the left side of the radio buttons. Checkbox labels might be placed on the right side of the check box.

[0086] In accordance with one embodiment of the invention, field numbering might be utilized. For example, a separate number might be used for each field across all forms. Such a numbering scheme might be beneficial for performance tuning. Further, such numbering might be beneficial for global changes to the forms of an insurance entity.

[0087] In accordance with one aspect of the invention, field lengths and sizes play a large role in the input of data by a data entry operator. Field lengths are the amount of characters permitted within a field. Field size is the actual size of the field, but may still be measured in characters. It may be the case for the “field length” to surpass the “field size.” In this situation, the additional characters will be stored for the field, but the additional characters won’t be visible without scrolling.

[0088] Field lengths should be large enough to handle all possible answers for a particular flow. The field should be able to accept values that may be not be the standard length for a field. The data entry operator should also be able to enter erroneous data, i.e., in conjunction with the

use of metadata to reflect that such entry is incorrect. The field size should only be large enough to show the most common field sizes. EXAMPLE: SSN is actually 11 characters in size, 9 characters for number and 2 characters for the dashes in the mask. EXAMPLE: The imaged form has a 12-digit SSN number erroneously entered, for a total of 14 characters including the dashes.

5 [0089] In the implementation of the system, various navigation buttons may be provided to the data entry operator. Figure 6 shows navigation buttons that may be provided on a data entry page. That is, Figure 6 shows the button name, one illustrative placement of the particular button, as well as action that results from selection of the particular button. Further, Figure 7 shows a basic example of manipulation of the form navigation buttons.

10 [0090] In accordance with one embodiment of the invention, the data collector tool 300 includes a “general comment” flow tool. A general comments flow is an individual flow that is associated to a particular form. This type of flow should be accessible to the data entry operator at anytime during the data entry process for its associated form. The purposes of general comments flow is to capture randomly entered information found on the imaged form, i.e., on the  
15 submitted form.

[0091] In accordance with one embodiment of the invention, “group boxes” may be used to group elements within a form flow. A title might be applied to the group box. Common usage for group boxes might include (1) to show sections within a form’ and (2) to group progressively disclosed or repeating groups of elements.

20 [0092] In accordance with one embodiment of the invention, masking may be used. Masks provide many benefits to the data entry process. Masks simplify data entry by eliminating the need for the DEO to enter commonly used characters. For example, a mask may simply provide a template to accept entered data 17. Masks also standardize the data being entered such that the



data can be structured for processing. Masks should not restrict the data being entered. Rather, masks should provide guidance for data entry. A field mask may be placed within a text box as desired. This will enable the DEO to see the characters populate the field in the correct mask.

Examples of masks which might be used are described below.

5 [0093] In accordance with one embodiment and as described above, the invention provides a tool which may be characterized as a “content sensitive” tool. The “content sensitive” tool provides the ability to dynamically change the data collector flow based on user selections, e.g., selections made by the data entry operator. Content sensitive logic gives the ability to show the DEO fields on a “need to see” basis. Such arrangement significantly reduces keystrokes and  
10 reduces clutter on the screen, and as a result, increases the efficiency of the data entry process and reduces opportunities for human error.

[0094] Content sensitive logic is driven from the selection of buttons, checkboxes, radio buttons, and combo and/or list boxes, for example. Content sensitive logic can be one or multiple flows driven by a single question. Multiple flows give the ability to provide multiple  
15 paths or structures for data entry. Figure 8 is a table showing examples of content sensitive logic in accordance with one embodiment of the invention.

[0095] In accordance with one embodiment of the invention, in the content sensitive processing, the data collector tool retains data in memory in a beneficial manner. To explain, data entered in a field or group (generated as a result of form flows and/or content sensitive  
20 logic) may be at jeopardy of loss if the DEO changes the value that disclosed the field(s). For example, the DEO might select an “other” value in a combo box that disclosed an “other” field and then enters data into that field. The DEO might then change the value in the combo box,

which now hides the “other” field. The DEO then might click back on the “other” value in the list. The data entered in the “other” text box should not be gone.

[0096] Accordingly, the data collector tool retains such data in memory. That is, data entered in progressively disclosed fields are retained in memory by the data collector tool. For example, the DEO selects a value in a combo box that disclosed several fields and then enters data into those fields. The DEO then changes the value in the combo box, which now discloses several new fields and the previously entered fields disappear. The DEO can then click back on the original selection to find that originally entered data is still intact. In accordance with one embodiment of the invention, this is necessary to ensure that a stray mouse-click does not accidentally cause significant data loss. Such arrangement also gives DEOs the ability to see fields, generated as a result of the content sensitive logic, from other selections without losing data.

[0097] However, it should be appreciated that data will indeed only be retained for a limited time. For example, in one embodiment, if the user leaves the current page, the previously entered information will not be available should the user return to that page.

[0098] In accordance with one aspect of the invention, a “remarks” or “details” flow might be utilized. Remarks and Details flows are used within the data collector tool to capture information provided within Remarks and Details sections of forms. For example, a “Remarks” flow may be used to capture information that is specific to an individual question as intended by the form. Alternatively, any information that is not associated with a specific question is to be captured in “general comments,” for example. As noted above, “remarks” and “details” may be used in the practice of the invention. In accordance with one aspect of the invention, “details” might be specifically for a particular question and may contain structured data elements. On the

other hand, “remarks” constitute a free-form place in which the applicant provides additional free-text information, as they desire.

[0099] As can be appreciated, information specific to a question on a submitted form may be found in several locations on a submitted form. For example, the DEO could expect to find

5 “Remarks” information in a combination of the following places on a submitted form, including circled, underlined, or otherwise altered text on a question, comments next to a question, text within the Details or Remarks sections of the imaged form, and/or text within the Details or Remarks overflow pages of the imaged form.

[00100] In accordance with one embodiment of the invention, all questions with an associated

10 Remarks flow will have a Remarks or Details button. These buttons serve the same purpose of initiating a Remarks flow. The particular name that the button is called may be coined in some logical manner, i.e., so as to be easily associated with the corresponding information on the submitted form 14, for example. To provide flexibility and increased speed of data entry, users, e.g., a data entry operator, can click the details or Remarks button next to a question to

15 immediately access the Remarks flow for a particular question. Figure 9 is a diagram 902 showing an example of a question with a Remarks flow.

[00101] In accordance with one embodiment of the invention, each Remarks flow will have an associate metadata called the “Remarks Flag.” This attribute identifies Remarks flows for a

question as being either true or false. True means that remark content for a question should exist,

20 which means that a Remarks flow and rules will be required for that Remarks flow. The default value for the flag is false. The flag may be set to “true” based on the following criteria, including (1) the DEO checks “Yes” or any other answer (i.e. “Totally Stopped Smoking”) that requires an associated Remarks flow; (2) selecting a Remarks or Details button, and/or (3) selecting from the

list of Remarks within the Remarks page. A user can reset this flag to “false” and remove all data within the remarks flow by selecting the remove button within each Remarks flow, in accordance with one illustrative embodiment of the invention.

**[00102]** Figure 10 shows a sample remarks page 102. The remarks page will be dynamically

created for all flows with a Remarks flag value of true. These Remarks flows will be sorted in the same order as the questions on the imaged form. In addition the Remarks Page will contain a list box for all questions with an associated Remarks flow. The DEO can select from this list to add additional Remarks flows to the Remarks Page. Added Remarks flows will be inserted into the sorted questions in the Remarks page. The purpose of this is to capture Remarks for questions that had a “No” answer, but still provided Remarks in the Remarks section for the question. (For Example: a user answers no to the question, however, they feel it is necessary to enter a remark explaining their no answer). The drop down should appear again so the user can enter possible additional Remarks.

**[00103]** In accordance with one embodiment of the invention, a “repeat” tool 1102 may also

be used, as shown in Figure 11. Repeats are used within the data collector tool to capture repeated information under some circumstances. In accordance with one embodiment of the invention, a “MORE” button on the particular form indicates a “repeat” option. For example, the DEO clicks the MORE button to repeat a flow within a group box. All repeated flows may be provided with a REMOVE button to delete the repeated flow. To explain, two people might be listed on a submitted form. This information should be data entered, i.e., “data entered” meaning that the data should be entered. However, the electronic form only has room for one person’s name. The “more” button is used by the data entry operator to repeat the name flow for that person.

[00104] It should be appreciated that a variety of screen layouts might be utilized in implementation of the data collector tool. The particular format of the screen layouts will of course depend on particular system requirements, as well as preferences of the intended users. Figure 12 illustrates a screen layout 1202 in accordance with one embodiment of the invention.

5 [00105] In accordance with one embodiment of the invention, signature verification may be utilized. Signature verification may be used to determine the legal validity of an application. Both the DEO and the exception manager (EM) may perform signature verification.

[00106] The DEO may perform an initial signature verification, as shown in the user interface 1302 of Figure 13. This initial verification might be performed by the data entry operator  
10 comparing the signature on the submitted form to the name on the submitted form. The purpose of this review is to ensure that (1) the document is signed, and (2) the signature matches the printed name on the document.

[00107] Further, the exception manager (EM) may perform a second pass of the verification when they receive a manual task. To clarify, the data collector tool 300 may perform some tasks  
15 automatically as a result of a deficient submitted form 15. For example, the data collector tool 300 might forward a supplemental form to the insurance applicant 10. On the other hand, some tasks may be manual and designated for the EM, e.g. such as the signature verification. That is, the EM may receive samples of tasks to perform signature verification. The purpose of this review is to validate the authenticity of signatures. Specifically, the EM may check whether the  
20 signature on different parts of a submitted form (or on different forms) match. The intent is to detect fraudulent signatures prior to issuance of the policy to the applicant. In accordance with one embodiment of the invention, this verification will appear as a task in an EM queue. The

User Interface will present a side-by-side representation of signature images for visual comparison.

**[00108]** DEO signature verification may be performed at a detailed level and a normal level.

The level of verification required is based on the current practice. The signature verification

5 might be characterized as either a detailed verification or a normal verification. A “detailed” verification is necessary for signatures that significantly impact the legal validity of the policy.

Alternatively, in some cases, a quick check to determine if a signature is present is sufficient.

This is considered a “normal” verification. Figure 13 is a user interface showing examples of both a detailed and a normal signature verification user interface.

10 **[00109]** In accordance with a further aspect of the invention, various validations may be performed during operation of the data collector tool, and in particular by the validation portion 330 of the data collector tool 300. For example, the data collector tool might perform what is characterized as a front-end validation, as well as a back-end validation. The front-end validation occurs during the data entry process within the data collector tool. In contrast, the back-end

15 validation occurs during rules processing. The purpose of front end validation is performed to enforce data integrity so that the rules engine can process the data. Both front-end validation and back-end validation uses metadata, i.e., attributes of the entered data, for validating forms. In one aspect, validation forces the DEO to either correct a field or set attributes for a field, i.e., reflecting incorrect entered data. Validation might be performed by the processing system at the

20 field level and/or the page level, as is described below.

**[00110]** With the above described aspects of the processing system in mind, hereinafter further details of the use of “attributes” of the entered data will be described, i.e., the use of metadata will be described.

[00111] In accordance with one embodiment of the invention, each form flow field should have the ability for the user to set field level attributes. These attributes are stored as field specific metadata in a suitable database. The processing of the entered data (i.e., processing of the internal electronic form) in the rules processing portion are dependent on the content of fields and their associated metadata. The details of a variety of attributes are described below with reference to the table 1402 of Figure 14.

[00112] Radio buttons and check boxes may be set at the question group level. This may require that the particular metadata apply to the question, and not answers to the question. The user can set this attribute while tabbed into any of the radio buttons or check boxes within the question, or in some other suitable manner. In particular, Figure 14 shows various attributes, i.e., metadata, that may be used in operation of the data collector tool. Further, Figure 14 shows examples of how the respective attributes might be activated. Lastly, Figure 14 shows the manner of identification of each respective attribute, i.e., with the use of font attributes. Further aspects of Figure 14, and the attributes set forth therein, are described below.

[00113] One attribute is metadata to show an altered field. This attribute identifies fields on the imaged form that have been altered in some way. The attribute can be applied to all fields in a form or groupings of checkboxes, for example.

[00114] Another attribute is metadata to show a blank field. This attribute identifies fields on the imaged form that have been intentionally left blank. The attribute can be applied to all fields or groupings of checkboxes in a form. Blank field attributes may set by a manual hot key selection, for example, or other suitable user interface option.

[00115] Another attribute is metadata to show an unreadable field. This attribute identifies fields on the imaged form that are unreadable by the DEO. The attribute can be applied to all

fields in a form. Unreadable field attributes are set by a manual hot key selection or other suitable user interface option.

[00116] Figure 14 also shows various other types or metadata described below. As should be appreciated, such metadata may be activated in any suitable manner, as is desired. Also, as shown in Figure 14, various font variations (including colors) are used to identify a particular attribute. However, any suitable identification might alternatively be used to identify the metadata.

[00117] In accordance with one aspect of the invention, it should be appreciated that metadata might be needed by the data entry operator, for example, for a particular field or for a group of fields. Illustratively, checkbox groupings might be defined according to business needs, as desired. In some cases, it may be desired that metadata be applied to each individual checkbox. On the other hand, other occasions may require metadata at the checkbox group level. In accordance with one embodiment of the invention, by default, metadata attributes are applied to groupings of checkboxes. Given this default, the data entry operator, for example, should specifically indicate on the form flow any checkboxes that should have metadata applied directly to the checkbox, rather than the grouping. Relatedly, such individual checkboxes would need to be provided with their respective labels.

[00118] It should be appreciated that it might be the situation that a field may be provided that is not specifically identified with a label. As described above, the particular font attributes of the field label is what conveys the attributes of the entered data. For example, fields without labels sometimes occur when they are structured in a table type format. However, entered data should have the capability to be associated with metadata. Various approaches might be used to



associate entered information with metadata. The particular approach to capture the metadata should be reflective of the particular business need.

**[00119]** For example, metadata might be applied to the row level. If no row label is available, suitable indications, such as a “\*” might be used as a row label. On the other hand, if row level

5 labeling is not possible, the metadata may be applied to the field level, i.e., a label might be associated with the field level. Labels may be created next to each field to which metadata is applied. Further, an “(A), (B), (...)” may be added to each field or grouping of fields within a row. Further, page level validation should have more than an “\*” displayed for each field, in accordance with one embodiment of the invention. The developer might create the label in the

10 page validation based on a combination of table name and row/column information for each field, for example. Labels associated with fields may be created as is desired or necessary.

**[00120]** With further reference to Figure 14, another type of attribute is “unreadable data.”

For example, unreadable data might indicate a Failed Checkbox Combination. To explain, a failed combination occurs when groups of checkboxes are set incorrectly. Failed combinations  
15 are identified with the Unreadable Attribute. For example, the DEO might select both Male and Female check boxes for a gender question. The DEO or the UI will identify a failed combination by setting the checkbox group as unreadable. This will occur in all groups of check boxes unless the groups are specifically set as multi-select.

**[00121]** Another attribute is failed mask fields. This attribute identifies fields on the data

20 collection form that have failed the masking for a field. This attribute may only be applied to text fields with masks and may be set only by the system.

**[00122]** Further, as shown in Figure 14, metadata may be provided that is reflective of multiples types of metadata, as described above. That is, an attribute might be “altered and

blank”; altered and unreadable; and/or altered with a failed mask. These attributes may be activated separately, or collectively, in some suitable manner.

[00123] Hereinafter, aspects of data validation will be described. In accordance with one embodiment of the invention, field level validation may be utilized. Field level validation applies to blank/unreadable fields, masked fields, and failed checkbox combinations, in accordance with one embodiment of the invention.

[00124] With regard to Blank/Unreadable Fields, in accordance with one embodiment of the invention, blank and unreadable fields are validated immediately upon exiting a field with a blank or unreadable attribute. Figure 15 is a table 1502 that describes conditions in which a validation query is presented to the data entry operator or other user due to contradictory entries. That is, for example, Figure 15 shows a particular condition that will prompt a validation query, as well as an illustrative message conveying the validation query.

[00125] Validation may also be performed on masked fields. That is, masked fields are validated immediately upon exiting a field with a mask. In accordance with one embodiment of the invention, a popup window will appear upon exiting a field that failed its mask validation. Figure 16 is a user interface 1602 that shows a generic mask error query to a data entry operator. Further, Figure 17 is a table 1702 that lists illustrative masks, which may be used in operation of the data collector tool 300.

[00126] Further, in accordance with one embodiment of the invention, failed Checkbox Combinations are validated immediately upon exiting a grouping of checkboxes that fails a combination check. Failed combinations occur in groupings of checkboxes where only one check box should be selected. A popup window will appear upon exiting a field with a failed combination. Figure 18 is a user interface 1802 that describes a particular condition that a field

has “failed the checkbox combination” and the resulting “failed combination” validation window.

[00127] In accordance with one embodiment of the invention, page level validation may be performed, as illustrated by the user interface 1902 of Figure 19. Page level validation occurs as a result of any action resulting in the exit of a page (i.e. submit, next, close, etc.), i.e., page level validation occurs whenever a page is exited. The data collector tool, and particularly the validation portion 330, may perform various checks of a page upon exiting from that page. Illustratively, the data collector tool will pop up a window listing all fields of a page (that fail a validation check) having a condition of an empty field (including drop lists with the default blank value), with no blank or unreadable attribute set and that have not been data entered, i.e. no characters. Another condition that will result in page validation failure is: a failed combination of checkboxes with no unreadable attribute set. A further condition that will result in page validation failure is: a failed address lookup, i.e., when the data collector tool could not validate a given address based on external databases, for example.

[00128] In accordance with one embodiment of the invention, the page level validation window 1902 forces the DEO to apply attributes to fields that fail a validation check. The DEO can edit blank and unreadable attributes for each field in the “Failed Field Validation Check” grouping. In accordance with one embodiment of the invention, all fields in page level validation default to the “Blank” attribute.

[00129] In accordance with one embodiment of the invention, the page level validation window will also dynamically include a section called “Failed Address Check” in the situation where a failed address is detected, as noted above. In this situation, the data entry operator will have the ability to select from a list of generated addresses based on current address data. The

DEO will have the ability to manually (via hotkey) set the address fields as blank or unreadable if the DEO can't find a suitable address from the generated list.

**[00130]** Figure 19 shows an example of a page level validation window. Upon completing each field the DEO will press UPDATE. The system will then process one of the following conditional requests. (1) If the DEO set an attribute for all fields, then update the form and process the original request. (2) If the DEO missed a field intentionally or mistakenly, then update the form and return the DEO to the page for completion of missed fields. This will allow the DEO to return to the form to fill in missed fields. In accordance with one embodiment of the invention, the form will hi-light all fields not updated by the DEO. In this manner, the DEO can see which fields she still needs to complete.

**[00131]** As described herein, various form flows may be utilized. In accordance with one embodiment of the invention, the form flows of the invention use visual representations of standard windows user interface components that are used for modeling the actual data entry forms. The method consolidates the presentation of form flow, form content, and field level information into a single easy to use document. Further to the description above describing various features of the form flow, Figure 20 is a sample form flow showing a blend of possible information found on a form flow, as well as descriptive text highlighting the features of the form.

**[00132]** Hereinafter, aspects of general implementation of the various embodiments will be described. Various system components and flowcharts are described above. The system of the invention or portions of the system of the invention may be in the form of a "processing machine," such as a general purpose computer, for example. As used herein, the term "processing machine" is to be understood to include at least one processor that uses at least one

memory. The at least one memory stores a set of instructions. The instructions may be either permanently or temporarily stored in the memory or memories of the processing machine. The processor executes the instructions that are stored in the memory or memories in order to process data. The set of instructions may include various instructions that perform a particular task or tasks, such as those tasks described above in the flowcharts. Such a set of instructions for performing a particular task may be characterized as a program, software program, or simply software.

[00133] As noted above, the processing machine executes the instructions that are stored in the memory or memories to process data. This processing of data may be in response to commands by a user or users of the processing machine, in response to previous processing, in response to a request by another processing machine and/or any other input, for example.

[00134] As noted above, the processing machine used to implement the invention may be a general purpose computer. However, the processing machine described above may also utilize any of a wide variety of other technologies including a special purpose computer, a computer system including a microcomputer, mini-computer or mainframe for example, a programmed microprocessor, a micro-controller, a peripheral integrated circuit element, a CSIC (Customer Specific Integrated Circuit) or ASIC (Application Specific Integrated Circuit) or other integrated circuit, a logic circuit, a digital signal processor, a programmable logic device such as a FPGA, PLD, PLA or PAL, or any other device or arrangement of devices that is capable of implementing the steps of the process of the invention.

[00135] It is appreciated that in order to practice the method of the invention as described above, it is not necessary that the processors and/or the memories of the processing machine be physically located in the same geographical place. That is, each of the processors and the

memories used in the invention may be located in geographically distinct locations and connected so as to communicate in any suitable manner. Additionally, it is appreciated that each of the processor and/or the memory may be composed of different physical pieces of equipment. Accordingly, it is not necessary that the processor be one single piece of equipment in one location and that the memory be another single piece of equipment in another location. That is, it is contemplated that the processor may be two pieces of equipment in two different physical locations. The two distinct pieces of equipment may be connected in any suitable manner. Additionally, the memory may include two or more portions of memory in two or more physical locations.

[00136] To explain further, processing as described above is performed by various components and various memories. However, it is appreciated that the processing performed by two distinct components as described above may, in accordance with a further embodiment of the invention, be performed by a single component. Further, the processing performed by one distinct component as described above may be performed by two distinct components. In a similar manner, the memory storage performed by two distinct memory portions as described above may, in accordance with a further embodiment of the invention, be performed by a single memory portion. Further, the memory storage performed by one distinct memory portion as described above may be performed by two memory portions.

[00137] Further, various technologies may be used to provide communication between the various processors and/or memories, as well as to allow the processors and/or the memories of the invention to communicate with any other entity; i.e., so as to obtain further instructions or to access and use remote memory stores, for example. Such technologies used to provide such communication might include a network, the Internet, Intranet, Extranet, LAN, an Ethernet, or

any client server system that provides communication, for example. Such communications technologies may use any suitable protocol such as TCP/IP, UDP, or OSI, for example.

**[00138]** As described above, a set of instructions is used in the processing of the invention.

The set of instructions may be in the form of a program or software. The software may be in the form of system software or application software, for example. The software might also be in the form of a collection of separate programs, a program module within a larger program, or a portion of a program module, for example. The software used might also include modular programming in the form of object oriented programming. The software tells the processing machine what to do with the data being processed.

**[00139]** Further, it is appreciated that the instructions or set of instructions used in the implementation and operation of the invention may be in a suitable form such that the processing machine may read the instructions. For example, the instructions that form a program may be in the form of a suitable programming language, which is converted to machine language or object code to allow the processor or processors to read the instructions. That is, written lines of programming code or source code, in a particular programming language, are converted to machine language using a compiler, assembler or interpreter. The machine language is binary coded machine instructions that are specific to a particular type of processing machine, i.e., to a particular type of computer, for example. The computer understands the machine language.

**[00140]** Any suitable programming language may be used in accordance with the various

embodiments of the invention. Illustratively, the programming language used may include assembly language, Ada, APL, Basic, C, C++, COBOL, dBase, Forth, Fortran, Java, Modula-2, Pascal, Prolog, REXX, Visual Basic, and/or JavaScript, for example. Further, it is not necessary that a single type of instructions or single programming language be utilized in conjunction with

the operation of the system and method of the invention. Rather, any number of different programming languages may be utilized as is necessary or desirable.

[00141] Also, the instructions and/or data used in the practice of the invention may utilize any compression or encryption technique or algorithm, as may be desired. An encryption module might be used to encrypt data. Further, files or other data may be decrypted using a suitable decryption module, for example.

[00142] As described above, the invention may illustratively be embodied in the form of a processing machine, including a computer or computer system, for example, that includes at least one memory. It is to be appreciated that the set of instructions, i.e., the software for example, that enables the computer operating system to perform the operations described above may be contained on any of a wide variety of media or medium, as desired. Further, the data that is processed by the set of instructions might also be contained on any of a wide variety of media or medium. That is, the particular medium, i.e., the memory in the processing machine, utilized to hold the set of instructions and/or the data used in the invention may take on any of a variety of physical forms or transmissions, for example. Illustratively, the medium may be in the form of paper, paper transparencies, a compact disk, a DVD, an integrated circuit, a hard disk, a floppy disk, an optical disk, a magnetic tape, a RAM, a ROM, a PROM, a EPROM, a wire, a cable, a fiber, communications channel, a satellite transmissions or other remote transmission, as well as any other medium or source of data that may be read by the processors of the invention.

[00143] Further, the memory or memories used in the processing machine that implements the invention may be in any of a wide variety of forms to allow the memory to hold instructions, data, or other information, as is desired. Thus, the memory might be in the form of a database to



hold data. The database might use any desired arrangement of files such as a flat file arrangement or a relational database arrangement, for example.

[00144] In the system and method of the invention, a variety of “user interfaces” may be utilized to allow a user to interface with the processing machine or machines that are used to implement the invention. As used herein, a user interface includes any hardware, software, or combination of hardware and software used by the processing machine that allows a user to interact with the processing machine. A user interface may be in the form of a dialogue screen for example. A user interface may also include any of a mouse, touch screen, keyboard, voice reader, voice recognizer, dialogue screen, menu box, list, checkbox, toggle switch, a pushbutton or any other device that allows a user to receive information regarding the operation of the processing machine as it processes a set of instructions and/or provide the processing machine with information. Accordingly, the user interface is any device that provides communication between a user and a processing machine. The information provided by the user to the processing machine through the user interface may be in the form of a command, a selection of data, or some other input, for example.

[00145] As discussed above, a user interface is utilized by the processing machine that performs a set of instructions such that the processing machine processes data for a user. The user interface is typically used by the processing machine for interacting with a user either to convey information or receive information from the user. However, it should be appreciated that in accordance with some embodiments of the system and method of the invention, it is not necessary that a human user actually interact with a user interface used by the processing machine of the invention. Rather, it is contemplated that the user interface of the invention might interact, i.e., convey and receive information, with another processing machine, rather than

a human user. Accordingly, the other processing machine might be characterized as a user.

Further, it is contemplated that a user interface utilized in the system and method of the invention may interact partially with another processing machine or processing machines, while also interacting partially with a human user.

5   **[00146]**   It will be readily understood by those persons skilled in the art that the present invention is susceptible to broad utility and application. Many embodiments and adaptations of the present invention other than those herein described, as well as many variations, modifications and equivalent arrangements, will be apparent from or reasonably suggested by the present invention and foregoing description thereof, without departing from the substance or scope of the  
10   invention.

**[00147]**   Accordingly, while the present invention has been described here in detail in relation to its exemplary embodiments, it is to be understood that this disclosure is only illustrative and exemplary of the present invention and is made to provide an enabling disclosure of the invention. Accordingly, the foregoing disclosure is not intended to be construed or to limit the  
15   present invention or otherwise to exclude any other such embodiments, adaptations, variations, modifications and equivalent arrangements.